

Pergunta 1

1.1

Inicialmente foi-nos pedido para executar 3 comandos semelhantes onde a única diferença era o parâmetro do número de bytes que eram postos no output. Estes comandos que seguem a forma "head -c XXXX /dev/random | openssl enc -base64" (onde XXXX é o número de bytes que se pretende), tem como função ir à "pool de entropia" /dev/random pegar em XXXX bytes random e codifica-los em base64 usando o openssl. Inicialmente foram pedidos 32 bytes aleatórios, sendo que foram obtidos quase instantaneamente. De seguida foram pedidos 64 bytes, desta vez foi necessário esperar uns poucos segundos antes de obter os resultados. Finalmente, foram pedidos 1024 bytes, sendo que foi necessário esperar bastante tempo antes de serem devolvidos 1024 bytes aleatórios. Isto é devido ao facto de o /dev/random ser uma "pool de entropia" com objectivos de ser criptograficamente segura, isto é, tenta angariar a maior entropia do sistema possível antes de dar uma resposta. Isto leva a que o programa pare sempre que não tem entropia suficiente para fornecer o número de bytes requeridos pelo utilizador. Em alternativa ao /dev/random, existe o /dev/urandom ("unlimited" random). Este ficheiro tem também como objetivo ser uma "pool de entropia", mas sempre que não possui entropia suficiente, serão reutilizados bytes usados anteriormente, isto seja, não são "tão aleatórios" como os bytes originados pelo /dev/random. Isto leva a que o tempo de resposta seja significativamente mais rápido para o mesmo número de bytes pedidos ao /dev/random, sem que a segurança seja comprometida completamente.

```
joao@joao:~$ head -c 32 /dev/random | openssl enc -base64
sF/fkhGix7dCJtt73sOUQ3ZdQwLSN6pqxo8pGF5w/+w=
```

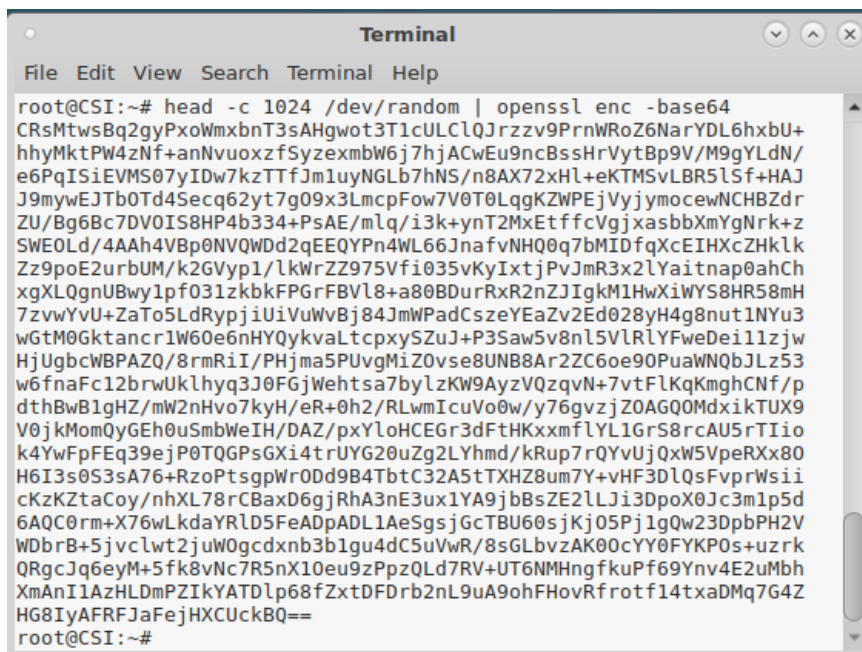
```
joao@joao:~$ head -c 64 /dev/random | openssl enc -base64
zgcniJNVA269Jr4UscVp3ZAgYx+sw5PEfXggTHg/g0oJq0STBrtkgNRzY8Fib/bg
vIz4Gs/z8H+yUwTRxYwUiQ==
```

```
joao@joao:~$ head -c 1024 /dev/random | openssl enc -base64
96/FKuf4RtECYmvXLCmxKrQG+3XmdENkynIj7BAG6fM9QQjxJKWmWuxkopdVNV5R
XFwCVK8Lbg6HZdTY05GUEmAC66i7xK+iQpAhOGZUnP/apFR09JuJy7PxGhep8w5Z
tRi8Bp7sB/tZZqR/bk9fvlPcF9LMZSv4mBP2kzWwLRx1C5oJkWaME28IGw+4Nc2
/5XYZf+28IyCCoiYMEYajBqm/YLj6mMCSgDzk1drY10lyLZs6gBHB/DBLRqbN81v
YoMXz0nebeDcreeQ/vCYJJMgGrv+ZwI8xpNeQUHBnt6FtQs0CrTex8U7A3aAEmtw
HJkoxAyKX5z36HEKTRghh0ji1NuZM+0PdSp3mSopgJetcUJ19Wxbfe0vOqhMcLK
HI9jtT62SLDnQYmuU/BkEa5LSn8sAtpdyXWKwCrH0gp0+lCBSxqC0CK5y8uP//5fH
chqi2rC67GqTmDcRl8qzjWQA4W5kLPHj3kBhft7RmQKRJxalULiZNEmpHigM0hoO
jdhHLh7Q9C7xTj0czT7ocEsVI0D2CBMcESrH2jr9H3hsIvIOf5GQ6Pqcgm8bgsBE
cF26vEmTbjwQYicXrkMtqSgvuE5LrZhBCR9AlsvoHm/eHaGu57Xl3nBSmQjw/uY3
2EuEUXXFemG4Zp+62CnJwGgft4e+1sXM5KdWmuRHmCzZbJ7EmhqHj1NR8znHF3uN
4KBK7LZR2sywor8mi++Y30g4CRBOPUpn/UUGWap2S/Y43vTPQ60osS1woIjAEbS8
XMMr//JXcENoARoDtOCRXXKoCi5GPr+wIgh2rlGgHoxcFQRGiAGStFeYgVnSEVxV
63wJ0KBVMBH4KwT0Eomkg94XNW0Ei0JhNOCDDJ2z/LPu9frMv8WV1qKuT0rGGvJ9V
hyn8SfR0f0r1aFmzcH20IPA3dX5UfaHVq3+QREE6Tt/IBBYlLXVm90SWdeTj200
x3iufSUQTnD3FC346ePUpQ+g9dA+byy3XQxkhHasN+IU01LpUWeVEsTFgrpBLorR
YpZ1zhL0qh/gGvOwcI+BzMvE63l0HatE8IpBVLXJoe1ih5UKzZBenNUNJxZCL9Wi
2WKf2sXqcrthi7jnJX2kInI4J90DZHb4G2IuWcTsa3kCCLHrEbhoA28uZUA0C0IX
ieRrvag3bNTDKkJKIFeVfoJ4hVtvaqSyHo1yR/LvBwEe4WsRKATMtjz/T9FVZHhK
22KMmDD+74j/iJ54bBfP6LT/eUjv3ex/KfLHe3XdoBkCduhJQ2yZxEyhZ4Pzgi7V
32nMkOXavwW5AfE0o2Zv0d2pghRVql/sJLIet3D5eq3hx5A1jxKnEp0TsFGp66PM
40RiIvHAUXyWLDWqHx9dKQ==
```

```
joao@joao:~$ head -c 1024 /dev/urandom | openssl enc -base64
0xi+ttEfqq30d5Wf3ePmXjNV0saMvgN3vug93g0HL56tsZ7YebEMT3hL5dq1/wb
0Pv4BdTJXpUS0evB0eojUxx7NbImYfq0U35La1BSMnDJY4GXgb5cZBbbD0u3HGr9
heAX15QSZ5B0vp6dmtB9osm8PewCRBPvfJtwRuunpWVmUYgHjgogpo+ngqXRgOG
XBxv78AtWZB+3RrCubGn1aCjKLogdB9XY6x2X1w0J1np1EIwk8khSNS+0iSSf2vF
ujA77YRphAJRVk1BwErHLVaYaj0kriHSwn+0g2rZp+Mssz2wWJf/D+6LR10thNLk
3nVKy+9qKSz3vPK7ac9zy9AiwsK1dM9dqAxbTtxKiX4AdDLchIgxoyKUG1fKa
xdMZCk50934Vv7cAc3IzpHKmK03EYgXbVbfY5HzrjEDAorKX5RuzvnyDTtwNABzf
b8DznZHRj1l1ZqyobNmY5KqCpCN6I60th5Ty02RxUFsALbKECffU8zxkLbS7DFHx
uWGYU/mn7lKqvT3SMjjU/b9vKjmrny+qrUERGknjtdHJfm8wtbsid8ZMoQOVGyX
XiKUFiTgbis3MIxiYmvRfn0j1Ubh4qIjmesZSb9jcKhSVAAMWx5x82+v+kHK//x
8yoh0SG+0B8/+6qujfUR7Knsi9umqWosFvLN21lJn6s1BZNyb5jw3j28uUdhzZQ0
mib/2Hc9uNffE5AZtQhQqmd8Ijgm3TiG5rdVQGGC+8RUwSaa6cl/hhwjEGS8uuUG
7vziJ/bXXkbjQUE6qK06FYBXaEAsdpv2yf2i6xZnV5QfW+Caumq+gPBEuw/PtCC
ec2wcJ8W/LnSPV/6AHEZ4eWR/2ddf1TLr/vBdtq47kLq8ZWdTNVnGqhQMCssZubY
asUusKJJnR3VgJL1fQ+4TpCme0U+SZ6XIKj/vHhWSJfMOHja1nos4d97gMWyGQpt
FD7zQBUEW627hxpjgWHUGcUSfYktpZx1rfYvQJ3kAi3gPj5l14BUwItPHsy2i2wK
zjLSWImQZfmdMUNFGL3J02vKzIrMlMKXh3jPMA0Uzw+44ypW29zT6V2sjpFML+xh
o1arZ2ZX6pGDxlrjkonwMih4pE0mQEESIHS/J0XlwAqcmiJl17WTWgAn5Ng8Rz4s
B319Z0pj7XlduUL0z1tKb/Fgh5meZrZB4VJHDLAVm6z2v74X1/gillhg55uteBW
fB5WhIscy9xY8qzovW0eC9VKNzqx24sMptUEaAbnv+0yvGLVUa8nye53vVU3BN9q
3x4VvVW3SGfmZ9t9jPjsJ0llzoL0ZqPPztIHDbJEvWHZ5yZ7oy+B0mjxKE6MrShN
KKvg8HqXvLaUM/i0+gqdsA==
```

1.2

Após a instalação do programa haveged voltamos a correr os comandos "head -c 1024 /dev/random | openssl enc -base64" e "head -c 1024 /dev/urandom | openssl enc -base64", sendo que desta vez os dois tiveram comportamentos semelhantes, ou seja, o tempo de execução foi drasticamente reduzido. Isto acontece pois o haveged tem como objectivo aumentar a entropia do sistema e desta forma o /dev/random nunca terá que ficar à espera de entropia suficiente para devolver os bytes, visto que o haveged está a gerar entropia.



```
Terminal
File Edit View Search Terminal Help
root@CSI:~# head -c 1024 /dev/random | openssl enc -base64
CRsMtwSbQ2gyPxoWmxbnT3sAHgwot3T1cULCLQJrzv9PrnWRoZ6NarYDL6hxbU+
hhyMktPw4zNf+anNvuoxzfSyzexmbW6j7hjACwEu9ncBssHrVytBp9V/M9gYLDN/
e6PqISiEVMS07yIDw7kzTTfJm1uyNGLb7hNS/n8AX72xHL+eKTMSvLBR5lSf+HAJ
J9mywEJTb0Td4Secq62yt7g09x3LmcpFow7V0T0LqgKZWPEjVyjymocewNCHBZdr
ZU/Bg6Bc7DVOIS8HP4b334+PsAE/mlq/i3k+ynT2MxEtffcVgjasbbXmYgNrK+z
SWEOLd/4AAh4VBp0NVQWdd2qEEQYPn4WL66JnafvNHQ0q7bMIDfqXcEIHxcZHKlk
Zz9poE2urbUM/k2GVyp1/lkWrZZ975Vfi035vKyIxtjPvJmR3x2LYaitnap0ahCh
xgXLQgnUBwylpf031zkbkFPGFrFBVl8+a80BDurRxR2nZJlGkM1HwXiWYS8HR58mH
7zvYvU+ZaTo5LdRypjiUiVuWvBj84JmWPAdCszeYEaZv2Ed028yH4g8nut1NYu3
wGtM0Gktancr1W60e6nHYQykvaLtcpxySZuJ+P3Saw5v8nl5VlRlYFweDei11zjw
HjUgbcWBPZQ/8rmRiI/PHjma5PUvgMiZ0vse8UNB8Ar2ZC6oe90PuaWNQbJLz53
w6fnaFc12brwUklhyq3J0FGjWehtsa7bylZKW9AyzVQzqvN+7vtFLKqKmgHCNf/p
dthBwB1gHZ/mW2nHvo7kyH/eR+0h2/RLwmIcuVo0w/y76gvzjZ0AG00MdxikTUX9
V0jkMomQyGEh0uSmbWeIH/DAZ/pxYloHCEGr3dFtHKxmfLYL1GrS8rcAU5rTIio
k4YwFpFEq39ejP0TQGPgXi4trUYG20uZg2LYhmd/kRup7rQYvUjQxW5VpeRXx80
H6I3s0S3sA76+RzoPtsgpWR0dd9B4TbtC32A5tTXHZ8um7Y+vHF3DLQsFvprWsi
cKzKZtaCoy/nhXL78rCBaxD6gjRhA3nE3ux1YA9jbBsZE2LLJi3DpoX0Jc3m1p5d
6AQc0rm+X76wLkdaYRLd5FeADpADL1AeSgsjGcTBU60sjKj05PjlgQw23DpbPH2V
WDbRb+5jvclwt2juW0gcdxb3b1gu4dC5uVwR/8sGLbvzAK00cYY0FYKP0s+uzrk
QRgcJq6eyM+5fk8vNc7R5nX10eu9zPpzQLd7RV+UT6NMHngfkuPf69Ynv4E2uMbh
XmAnI1AzHLDmPZIKYATDlp68fZxtDFDrb2nL9uA9ohFHovRfrotf14txaDMqG7G4Z
HG8IyAFRFJaFejHXCUCk8Q==
root@CSI:~#
```

```
Terminal
File Edit View Search Terminal Help

root@CSI:~# head -c 1024 /dev/urandom | openssl enc -base64
s0pb/2/IaDSLu0xqjn4KzTegpJkAeR+pWHdS0wvehVC8Zju/dyDuqn9BzqPL2AT8
9Gr0DT7TthRC6cvTK8R+xEoyR6L7a7XIprSMU83Kxja08Fr0o8ESItZrU8ZLfpAj
mYqPrGK2n+orWYXoXT2PFjZE4G50RVITyyHmTKXShHUDMA2NswSPbGP8WW9d+Mdm
4rVpuK1kTQpFt82PXGyFwQeAYJQuZ4xp5i+tQd/hZjDzbeqWREjpuWn6c+gUAs
c/Q/cTHP0IpX7p9D9+sJ+ejNlkfXRFU50RIZHFz8qpzXJh5uNydfxtMFq4ovtBt0
L0DYWPh8F5j1glJJZ6znwVqSsscTD7JpybT67oAQXRlvYQl+ml6kD/p4YgqW32mv
HC4xxjVdS/lrox500iHDGfGLir/XtD0LzpEKxZKVYyswnSuQq97dibLeTcPGIR9
VJZB/necIL5WY0szYJU6bsHzkapqcRdtH94oJR6+wbV698ZI4sZpjzP8RFzwqEVd
lfeMZ1neyiEPG3ILDpuEifw8E5W8oLYNyhKG+HHHxk4hP9GVzPH24vW9g07jN9Y
VgiV4H7pEdHrLkFrbyvQ4Tzt0bY5f3j0d7f6nYJyemNJTeTDHr97hVv5ztwYJjG
l56z1Zyd/C7SgtMNBhRAwG7SW8aX0k80NC5tpzaun0QpyE85xv1zyriu95MkIDP5
zagLQtjr+6xVKVQmAh0D1JsrliqzX569E8iuIiU42UlaICRbUlfoRz6sGXFangA
AvWb1REBEjJFkniedoQT8F58nzllRTJJHaJXEvIwujQL3hi7l3WPkq0LSCxKNEbo
/EST4SK9eJ4A+fRf6KMEFJ/c06En+laPCGzgjHSQqUt0x0Zs5gh2UkXhGA0Mq/L5
LABivzHCU0ZbyPZm58BaKS3dPs5haedp8Y6jvh70uRcvlVic0LVpm64WL3+D0cU
PN015jFyllHT04a0nDkrqueYjSS/buIMC/5wUiWkExhJSFBEVXUZuvA4iA/VRWt
fEduS/2XsujRV2g9tA05qZF94x/WTADUT3pvo6hJydzatbp86zEXXge0GfRwiJMp
kNtYFF39Lkc1o37/wF0RorGldCvKWM0Ea+10qQ6BLJIYJx4xNnKaxUW40kaG8V
OZ+d5orY8gaHHmTYRr5XpQPi2W0F1pf/vBR2i43Jv4mK/aMSjXqzrDg2ezwZf0rn
YyvBFIRdpzued99l3T9rvaPeB6Qmj7yPM8xUyU3PFbzGLDfRxLK50rvbENGWtwpI
XPIuLL3Huye/ZjiF2xNTXA1i9BS7DMsNXhPpIIqX8nGvI783qF4qaR0DPoYmPdkN
gT+8LYaSK5z/o0qV92JqhQ==
root@CSI:~#
```

1.3

Como é foi possível ver no output gerado pelo ficheiro generateSecret-app.py, o segredo apenas contem letras (maiúsculas e minúsculas) e dígitos. Isto deve-se ao facto de no módulo eVotUM.Cripto, quando se gera o segredo, tudo o que não é um dígito ou carácter ascii é rejeitado, isto é, não é adicionado ao segredo a ser devolvido. Este processo é repetido até o segredo a ser construído atingir o tamanho definido pelo utilizador.

Isto poderia ser contornado retirando esta condição do código, ou alternativamente adicionar exceções para os caracteres adicionais pretendidos no segredo.

```
def generateSecret(secretLength):
    """
    This function generates a random string with secretLength characters (ascii_letters and digits).
    Args:
        secretLength (int): number of characters of the string
    Returns:
        Random string with secretLength characters (ascii_letters and digits)
    """
    l = 0
    secret = ""
    while (l < secretLength):
        s = utils.generateRandomData(secretLength - l)
        for c in s:
            if (c in (string.ascii_letters + string.digits) and l < secretLength): # printable character
                l += 1
                secret += c
    return secret
```